

**本科生实验报告**

|  |  |
| --- | --- |
| **实验课程** | 《R语言程序设计》 |
| **学院名称** | 数理学院 |
| **专业名称** | 应用统计学 |
| **学生姓名** | 龙浩然 |
| **学生学号** | 202220020122 |
| **指导教师** | 徐松浦 |
| **实验地点** | 综实521 |
| **实验成绩** |  |

**二〇二四 年 九 月 —— 二〇二五 年 一 月**

**填写说明**

1. 适用于本科生所有的实验报告（印制实验报告册除外）；
2. 专业填写为专业全称，有专业方向的用小括号标明；
3. 格式要求：
4. 用A4纸双面打印（封面双面打印）或在A4大小纸上用蓝黑色水笔书写。
5. 打印排版：正文用宋体小四号，1.5倍行距，页边距采取默认形式（上下2.54cm，左右2.54cm，页眉1.5cm，页脚1.75cm）。字符间距为默认值（缩放100%，间距：标准）；页码用小五号字底端居中。
6. 具体要求：

**题目**（二号黑体居中）；

**摘要**（“摘要”二字用小二号黑体居中，隔行书写摘要的文字部分，小4号宋体）；

**关键词**（隔行顶格书写“关键词”三字，提炼3-5个关键词，用分号隔开，小4号黑体)；

正文部分采用三级标题；

**第1章** ××(小二号黑体居中，段前0.5行)

**1.1** ×××××小三号黑体×××××（段前、段后0.5行）

**1.1.1**小四号黑体（段前、段后0.5行）

**参考文献**（黑体小二号居中，段前0.5行），参考文献用五号宋体，参照《参考文献著录规则（GB/T 7714－2005）》。

实验题目 R语言扩展案例的修改与调试

# 实验内容

1.5、5.1.2考试成绩的回归分析

2.5.1寻找连续出现1的游程

2.5.2预测离散值时间序列

2.9.1 度量相关性

2.9.2对鲍鱼数据集重新编码

3.2.3图像操作

3.2.5 生成协方差矩阵

3.3.2寻找异常值

3.4.2 找到图中距离最近的一对端点

4.2.4、4.4.2文本词汇索引

4.4.3鲍鱼数据

12.1.4在一张图中绘制两条密度曲线

# 实验要求

1. 程序完整：必须包括题目、代码、运行结果、结果分析；
2. 对程序中主要代码、变量，必须加以注释（例如使用word 菜单中的“插入”——）“形状”命令），以示和其他人报告的区别；
3. 针对书上源程序，可以试着对主要变量的标识符名称、变量值等进行修改、调试。

# 实验准备与实验过程

## 实验环境与数据

1. 安装软件：R-4.0.2-win.exe和RStudio-1.2.1335.exe；
2. 题目要求和数据请参考教材《R语言编程艺术》麦特洛夫 (Norman Matloff) (作者),陈堰平(译者) 机械工业出版社; 第1版 (2013年5月1日)中的内容；

## 实验过程

（包括题目、修改后的代码、运行结果）

### 1.5、5.1.2考试成绩的回归分析

要求，更具期中成绩和平时测试，建立预测期末成绩的回归方程

代码：

exams <- read.table("C:/Users/13708/OneDrive/桌面/R语言/data/ExamsQuiz.txt", header = FALSE)

exams

lma <- lm(exams[,2] ~ exams[,1]+exams[,3])

lma$coef

运行结果：



得到回归方程为：

### 2.5.1寻找连续出现1的游程出现的索引

要求：寻找向量y (1,0,0,1,1,1,0,1,1)的连续出现1的游程

方法1:

findruns <-function(x,k){

n<-length(x)

runs<-NULL

for (i in 1:(n-k+1)){

if(all(x[i:(i+k-1)]==1))

runs<-c(runs,i)

}

return(runs)

}

y<-c(1,0,0,1,1,1,0,1,1)

findruns(y,3)

运行结果：



方法2：一开始给runs分配空间

findruns <-function(x,k){

n<-length(x)

runs<-vector(length=n)

count <- 0

for (i in 1:(n-k+1)){

if(all(x[i:(i+k-1)]==1)){

count <- count+1

runs[count]<- i

}

}

if(count>0){

runs <- runs[1:count]

}

else{

run <- NULL

}

return(runs)

}

y<-c(1,0,0,1,1,1,0,1,1)

findruns(y,2)

运行结果：



### 2.5.2预测离散值时间序列

要求：使用训练集中的数据，找到最好的K值，预测下雨的概率

方法1：

values <- sample(c(0, 1), size = 500, replace = TRUE, prob = c(0.6, 0.4))

predA <- function(x,k){

n <- length(x)

half <-k/2

prd <- vector(length = n-k)

for(i in 1:(n-k)){

if(sum(x[i:(i+(k-1))])>=half){

prd[i]<-1

}

else{

prd[i]<-0

}

}

return(mean(abs(prd-x[(k+1):n])))

}

point <- vector(length = 18)

for(i in 3:20){

point[i-2] <- predA(values,i)

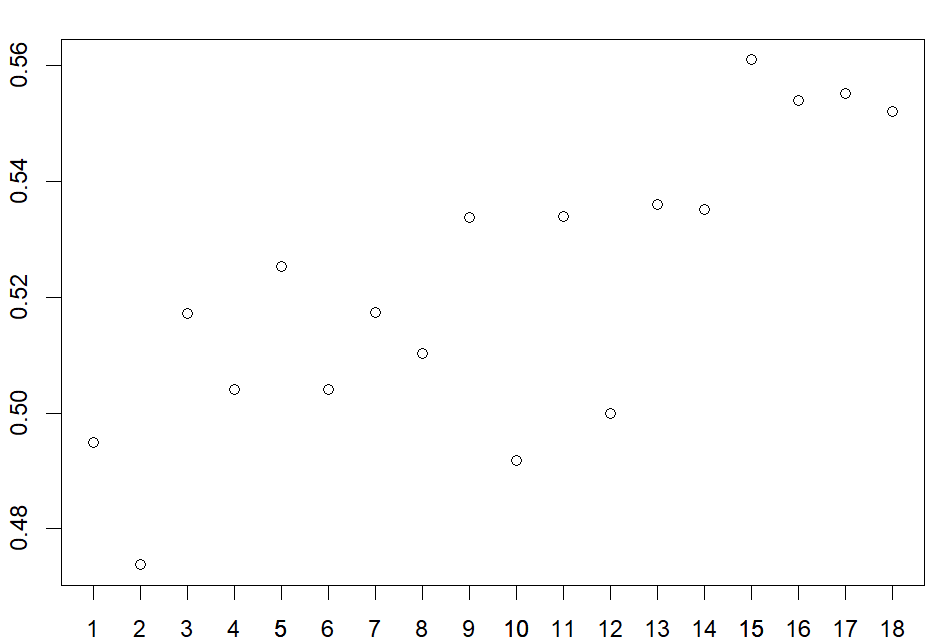
}

point

plot(c(1:18),point,xlim = c(1, 18))

axis(1, at = seq(1, 18, by = 1))

运行结果：



学习曲线图得到，K=15的时候预测结果最好，概率为0.51

### 2.9.1 度量相关性

要求：统计X和Y两个时间序列同时上升或下降的次数，来确定两个序列的相关性。

代码：

findud <- function(v){

vud <- v[-1] - v[-length(v)]

return(ifelse(vud>0,1,-1))

}

udcorr <- function(x,y){

ud <- lapply(list(x,y),findud)

return(mean(ud[[1]]==ud[[2]]))

}

x <- c(5,12,13,3,6,0,1,15,16,8,88)

y <- c(4,2,3,23,6,10,11,12,6,3,2)

udcorr(x,y)

运行结果：



### 2.9.2对鲍鱼数据集重新编码

要求：按照性别对鲍鱼的数据进行分组，并且绘制出散点图来描述鲍鱼的直径和长度的关系。

1. 部分鲍鱼的数据：

代码：

aba <- read.csv("C:/Users/13708/OneDrive/桌面/R语言/data/Abalone.data", header = TRUE, as.is = TRUE)

aba

grps <- list()

for(gen in c("M","F")){

grps[[gen]] <- which(aba[,1]==gen)

}

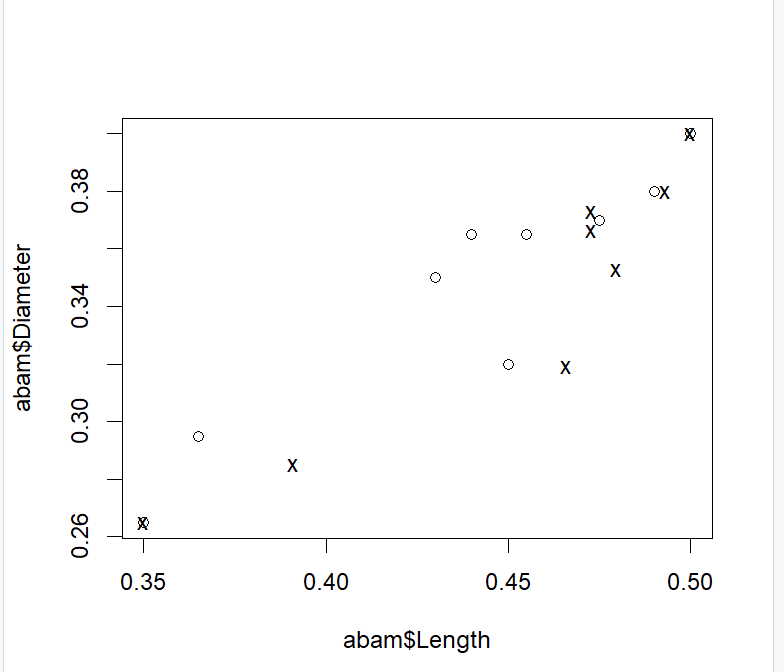
abam <- aba[grps$M,]

abaf <- aba[grps$F,]

plot(abam$Length, abam$Diameter)

par(new=TRUE)

plot(abaf$Length, abaf$Diameter, pch="x", axes=FALSE, xlab="", ylab="")



2.全部鲍鱼的数据

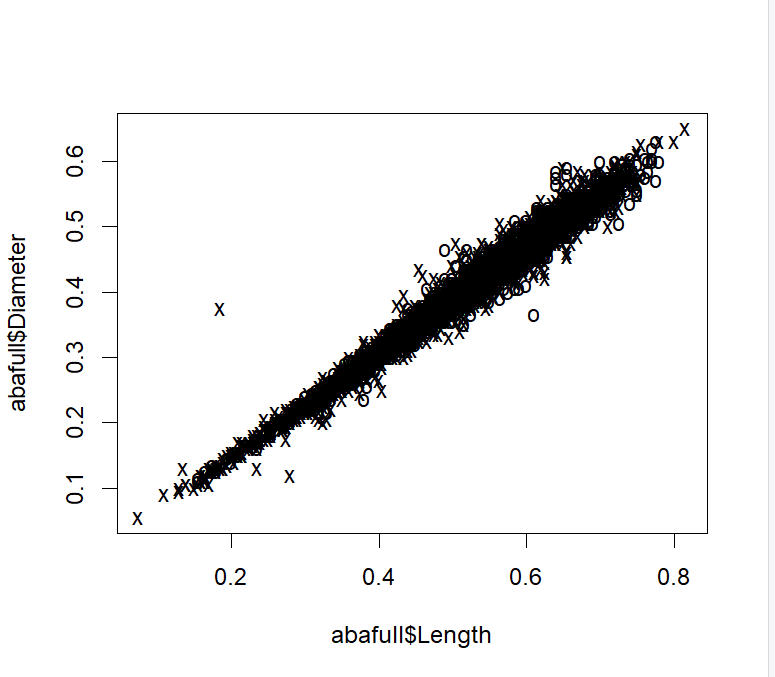
代码：

abafull <- read.csv("C:/Users/13708/OneDrive/桌面/R语言/data/Abalone\_0.data", header = TRUE, as.is = TRUE)

pchvec <- ifelse(abafull$Gender == "M","o","x")

plot(abafull$Length,abafull$Diameter,pch=pchvec)

运行结果：



### 3.2.3图像操作

要求：读取图像，同时修改图像内容

1. 用白色区块盖住罗斯福总统的脸

代码：

library(pixmap)

mtrush1 <- read.pnm("C:/Users/13708/AppData/Local/R/win-library/4.3/pixmap/pictures/mtrush1.pgm")

mtrush1

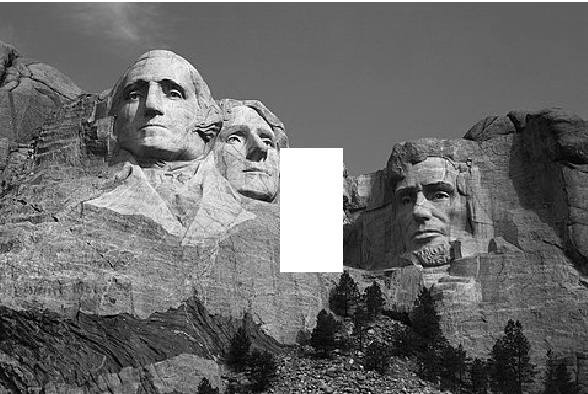
plot(mtrush1)

mtrush2 <- mtrush1

mtrush2@grey[120:220,230:280] <- 1

plot(mtrush2)

运行结果：



1. 在罗斯福总统的图层上面添加随机噪音

代码：

blurpart <- function(img, rows, cols, q) {

lrows <- length(rows)

lcols <- length(cols)

newimg <- img

randomnoise <- matrix(runif(lrows \* lcols), nrow = lrows, ncol = lcols)

newimg@grey[rows, cols] <- (1 - q) \* img@grey[rows, cols] + q \* randomnoise

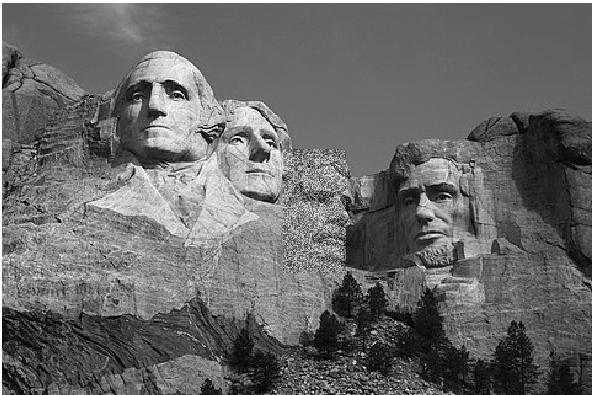
return(newimg)

}

mtrush3 <- blurpart(mtrush1, 120:220, 230:280, 0.5) #添加的噪音量为50%

plot(mtrush3)

运行结果：



### 3.2.5 生成协方差矩阵

要求：生成一个主对角线上面的元素为1，其他元素都为0.2的协方差矩阵

代码：makecov <- function(rho,n){

m <- matrix(nrow=n,ncol=n)

m <- ifelse(row(m)==col(m),1,rho)

return(m)

}

cov = makecov(0.2,3)

cov

运行结果：



### 3.3.2寻找异常值

要求：根据中位数，找出矩阵每行的异常值

代码：findols <- function(x){

findol <- function(xrow){

mdn <- median(xrow)

devs <- abs(xrow-mdn)

return(which.max(devs))

}

return(apply(x,1,findol))

}

x <- c(41.57113, 62.64265, 50.73139, 52.25710, 51.19273,

50.84775, 59.86869, 114.51419, 53.16677, 47.79379,

43.81002, 49.69344, 62.38947, 51.65549, 114.1919,

43.22145, 49.69344, 56.05712, 47.60135, 55.51238,

194.1145, 47.18273, 53.28691, 52.12064, 43.65536,

46.48138, 56.85044, 53.18482, 53.18482, 52.46300,

43.36258, 43.01319, 50.64394, 41.56041, 59.87370,

61.13737, 52.10061, 51.85978, 49.01487, 48.63963,

54.09298, 191.1145, 47.73394, 114.1919, 44.61585,

45.95224, 59.26103, 41.42285, 50.10780, 52.87698)

data <- matrix(x,nrow = 5, ncol = 10)

data

findols(data)

结果：返回了每行矩阵中，异常值的位置



### 3.4.2 找到图中距离最近的一对端点

要求：根据城市的距离矩阵，输出城市间最短的距离

代码：

mind <- function(d){

n <- nrow(d)

dd <- cbind(d,1:n)

wmins <- apply(dd[-n,],1,imin)

i <- which.min(wmins[2,])

j <- wmins[1,i]

return(c(d[i,j],i,j))

}

imin <- function(x){

lx <- length(x)

i <- x[lx]

j <- which.min(x[(i+1):(lx-1)])

k <- i+j

return(c(k,x[k]))

}

x <- c(0,12,13,8,20,12,0,15,28,88,13,15,0,6,9,8,28,6,0,33,20,88,9,33,0)

q <- matrix(x,nrow=5,ncol=5)

q

mind(q)

运行结果：



### 4.2.4、4.4.2文本词汇索引

要求1：统计文本中的单词出现的次数

代码：findwords <- function(tf){

txt <- scan(tf,"")

wl <- list()

for(i in 1:length(txt)){

wrd <- txt[i]

wl[[wrd]] <- c(wl[[wrd]],i)

}

return(wl)

}

findwords("C:/Users/13708/OneDrive/桌面/R语言/data/testconcord\_1.txt")

运行结果：（局部）



要求2：按照utf-8编码，排列单词同时显示出现的位置

代码：

alphaw <- function(wl){

nms <- names(wl)

sn <- sort(nms)

return(wl[sn])

}

alword <- alphaw(word)

运行结果：（局部）



### 4.4.3鲍鱼数据

使用lapply（）函数，统计鲍鱼数据向量中“M”,“F”,“I”，三种分类每种有多少。

代码：

g <- c("M","F","F","I","M","M","F")

lapply(c("M","F","I"),function(gender) which(g==gender))

运行结果：



### 12.1.4在一张图中绘制两条密度曲线

使用density函数，来计算密度函数的估计值，并且绘制在同一张图中

library(readxl)

testscore <- read\_excel("C:\\Users\\13708\\OneDrive\\桌面\\R语言\\data\\testscore.xlsx")

d1 = density(testscore$Exam1,from=0,to=100)

d2 = density(testscore$Exam2,from=0,to=100)

plot(d1,main="",xlab="")

lines(d2)

运行结果：

图表, 直方图

描述已自动生成

|  |  |
| --- | --- |
| **学生实验 心得** | 在本学期的《R语言实用编程》课程中，我深刻体会到 R 语言在数据分析中的强大功能与灵活性。通过一系列实验，我不仅巩固了理论知识，还提升了实践能力。  首先，在回归分析实验中，我学习到如何运用线性模型分析考试成绩的影响因素。这让我理解了数据建模的重要性，并激发了我对数据关系探索的兴趣。  在寻找连续出现1的游程和预测离散值时间序列的实验中，我掌握了数据处理和时间序列分析的基本技巧。尤其是时间序列预测，让我感受到数据的动态特性及其应用价值。  我也通过度量相关性和异常值检测实验，了解了数据清洗与特征工程的重要性，确保分析结果的可靠性。同时，图像操作和协方差矩阵生成的实验让我领悟到可视化与数据之间的密切联系。  文本词汇索引的实验让我意识到 R 语言在文本分析中的广泛应用，尤其是在信息检索和自然语言处理领域。最后，在同一图中绘制两条密度曲线的实践，帮助我更直观地比较不同数据分布。  总的来说，这门课程不仅让我提升了编程技能，更培养了我对数据分析的全面理解。我期待在未来的学习中，能够将这些技能运用到实际问题中，继续探索数据的奥秘。  学生（签名）：  2024年 10月 21 日 |
|  |  |

**指导教师评分：**

|  |  |  |  |
| --- | --- | --- | --- |
| **评 分 指 标** | | **分 值** | **得 分** |
| 实验过程考核  （20%） | 实验考勤率 | 10 |  |
| 按时完成度 | 10 |  |
| 实验报告成绩  （80%） | 逻辑正确性 | 10 |  |
| 结构合理性 | 10 |  |
| 内容详尽性 | 10 |  |
| 过程清晰性 | 10 |  |
| 结论合理性 | 10 |  |
| 格式规范性 | 10 |  |
| 语言通顺性 | 10 |  |
| 图表美观性 | 10 |  |
| 合 计 | | 100 |  |

教师签名：

2024年 11月 日